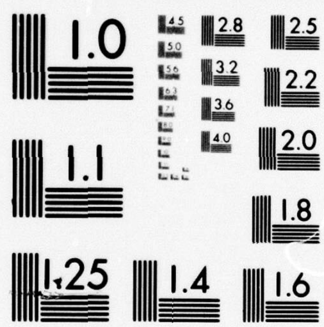
AD-A079 534 SOUTHEASTERN MASSACHUSETTS UNIV NORTH DARTMOUTH F/G 20/13
A COMPUTATIONAL METHOD FOR SOLVING PARABOLIC PARTIAL DIFFERENTI--ETC(U)
NOV 79 T B SHEN AFOSR-78-3538

UNCLASSIFIED AFOSR-TR-80-0017 NL

1 OF 1
AD A079534

END
DATE
FILMED
3-80
DDC



MICROCOPY RESOLUTION TEST CHART

PHOTOGRAPH THIS SHEET

ADA079534

DTIC ACCESSION NUMBER

II

LEVEL

INVENTORY

AFOSR-TR-80-0017

DOCUMENT IDENTIFICATION

Grant AFOSR-78-3538

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR		
NTIS	GRA&I	☒
DTIC	TAB	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION /		
AVAILABILITY CODES		
DIST	AVAIL AND/OR SPECIAL	
A		

DISTRIBUTION STAMP

DDC
RECEIVED
JAN 16 1980
D

DATE ACCESSIONED

80 1 14 040

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DTIC FORM OCT 79 70A

DOCUMENT PROCESSING SHEET

AFOSR-TR- 80-0017

A COMPUTATIONAL METHOD FOR SOLVING PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS

by

Thomas B. C. Shen
Associate Professor of Mechanical Engineering
Southeastern Massachusetts University

November 1979

Final Report

prepared for

Air Force Office of Scientific Research
under
Grant AFOSR - 78 - 3538

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: **AFOSR-TR- 80-0017**	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A COMPUTATIONAL METHOD FOR SOLVING PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS	5. TYPE OF REPORT & PERIOD COVERED: FINAL 01 JAN 78 - 31 JUL 79
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): THOMAS B. C. SHEN	8. CONTRACT OR GRANT NUMBER(s): AFOSR-78-3538
9. PERFORMING ORGANIZATION NAME AND ADDRESS: SOUTHEASTERN MASSACHUSETTS UNIVERSITY NORTH DARTMOUTH, MA 02747	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2308D9 61102F
11. CONTROLLING OFFICE NAME AND ADDRESS: AIR FORCE OFFICE OF SCIENTIFIC RESEARCH/NA BLDG 410 BOLLING AIR FORCE BASE DC 20332	12. REPORT DATE: NOV 79
	13. NUMBER OF PAGES: 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

NEW COMPUTATIONAL TECHNIQUES; RECURRENCY TECHNIQUE; PARABOLIC DIFFERENTIAL EQUATION SOLUTIONS; ORDINARY LINEAR AND NONLINEAR EQUATION SOLUTION; TURBULAR FLOW HEAT TRANSFER ANALYSIS; COAXIAL LAMINAR TURULAR FLOWS; RADIATIVE FIN ANALYSIS; HOT SURFACE FUEL SPRAY IGNITION AND COMBUSTION

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents a new numerical method, the error method, for solving parabolic type partial differential equations, linear or nonlinear. In particular, by comparing with the regular successive iterative method, more beneficial results in the application to nonlinear problems. Three nonlinear examples were studied by using this method. All resulted in large reduction of number of iteration loops and CPU time required in comparing to the corresponding regular successive method used. Generalization and modification of this method appears appropriate to extend its application to elliptical type partial differential

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

equation so that problems with "isolated" events (such as those with ignition spots) may be handled with this method.

TABLE OF CONTENTS

ABSTRACT

This paper presents a new numerical method, the error method, for solving parabolic type partial differential equations, linear or non-linear. In particular, by comparing with the regular successive iterative method, more beneficial results in the application to non-linear problems. Three non-linear examples were studied by using this method. All resulted in large reduction of number of iteration loops and CPU time required in comparing to the corresponding regular successive method used.

Generalization and modification of this method appears appropriate to extend its application to elliptical type partial differential equation so that problems with "isolated" events (such as those with ignition spots) may be handled with this method.

PREFACE

The financial support of Air Force Office of Scientific Research through Grant AFOSR-78-3538 is highly appreciated. Many thanks to the great patience of Dr. Bernard T. Wolfson who managed this program throughout the entire period.

I am grateful to the computer services provided by computer centers of both Harvard University and Southeastern Massachusetts University.

I. Introduction

This paper presents a new numerical method, the error method, for solving parabolic type partial differential equations, linear or non-linear. Observing from the cases studied, it appears that the application of the error method to non-linear problems results in more beneficial than to linear problems in comparing with the conventional workable successive iterative numerical scheme. This new method features wide applicability. Its computational scheme converges fast. The following sections will describe the fundamental derivation of this method, its applications to non-linear problems and the future potential developments.

II. The Successive Iterative Method

To construct the basic numerical scheme of the error method, let's start by reviewing the conventional successive iterative method. Consider the problem of an one-dimensional, unsteady heat conduction in a solid plane wall with finite thickness. Its governing equation is,

$$T_t = T_{xx} \quad , \qquad (1)$$

where T, the unknown temperature field and x,t, the effective spacious and time coordinates. The set of initial and boundary conditions may be,

$$\left.\begin{aligned} t = 0, &\quad T = T_o(x) \\ x = 0, &\quad T = C_1 \\ x = 1, &\quad T = C_2 \quad , \end{aligned}\right\} \qquad (2)$$

where l, the effective wall thickness and C_1, C_2, two fixed temperature values on the two wall surfaces. $T_0(x)$ represents the initial temperature profile across the wall thickness.

To such a problem, the finite difference successive iterative numerical method can be used. The field, along x direction, is first

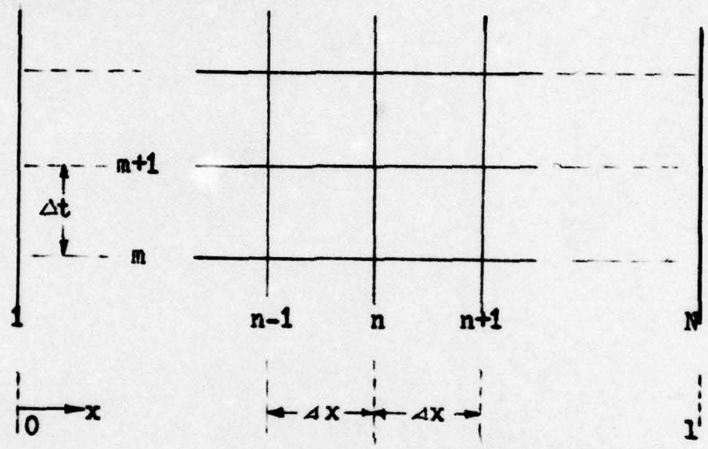


Figure 1. Grid System for the Numerical Scheme

divided into (N-1) grids. Each grid sizes Δx. An implicit finite difference equation is constructed from equation (1), i.e.

$$T_{m+1,n} = \frac{T_{m+1,n+1} + T_{m+1,n-1}}{2\left(1 + \frac{\Delta x^2}{\Delta t}\right)} + F(T_{m,*}) \quad , \qquad (3)$$

where $$F(T_{m,*}) \equiv \frac{T_{m,n+1} - 2T_{m,n} + T_{m,n-1}}{2\Delta x^2\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta t}\right)} + \frac{T_{m,n}}{\Delta t\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta t}\right)} ,$$

and n varying from 2 to N-1. The solution of equation (3), together with the two boundary conditions specified, determine the various $T_{m+1,*}$ as $T_{m,*}$ are

given. With successive iterative method, solution of equation (3) starts by assuming a set of $T_{m+1,*}$ values, together with the specified information for $F(T_{m,*})$ from time instant, m, substituting into the right side of equation (3), then calculating a new set of $T_{m+1,*}$. This new set of $T_{m+1,*}$ is then replacing the old set of $T_{m+1,*}$ to plug into the right side of equation (3) again and calculating the next new set of $T_{m+1,*}$. Such procedure repeats itself until the calculated set of $T_{m+1,*}$ revealing no significant difference from the previous set of $T_{m+1,*}$. The last set of calculated $T_{m+1,*}$ values are considered the satisfied solution. The selection of the starting $T^{(1)}_{m+1,*}$ profile is not unique. Ordinarily, to use the profile, $T_{m,*}$, as the first trial profile, $T^{(1)}_{m+1,*}$ appear not a bad choice. Since, intuitively, as we may expect that the correct unknown profile, $T_{m+1,*}$, *should not be different from* the known profile of the previous time instant, $T_{m,*}$, too much if Δt is small enough and no "isolated" event occurs. To assure the stable convergence of the calculation steps and lead to reasonable solution, it appears that the criterian,

$$\frac{\Delta x^2}{\Delta t} > 2 \qquad (4)$$

ought to be observed in selecting the relative sizes of Δx and Δt.

According to above description, a Fortran program (SPD1), attached as Appendix A, was constructed to allow digital computer to perform the steps in the iterative computation procedure for this one dimensional, unsteady heat conduction problem. A case with the initial condition, $T_0(x)$, possessing

one surface temperature at elevated level was studied. Some calculated temperature profiles at various time instants are shown in Table 1. In this table, the number of iteration loop required for calculating each temperature profile is also indicated; so is the total CPU time used. Also, cases with highly non-linear initial conditions were also investigated with this program, the results indicate that the successive method appears very effective to handle such linear problems.

Table 1 Some Temperature Profiles for a One-Dimensional, Unsteady Heat Conduction Problem (Successive Method)

Initial Temperature Profile: T=200.0 at x=0.0 and t=0.0
T= 40.0 at x 0.0 and t=0.0
Boundary Conditions: T=200.0 at x=0.0 and t 0.0
T= 40.0 at x=1.0 and t 0.0
The Label: Orderly number of Temp Profile, No. of Iteration Loop, Time Inst. Values of Temperature Profile

	1	0	0.00000		
200.00000	40.00000	40.00000	40.00000	40.00000	40.00000
40.00000	40.00000	40.00000	40.00000	40.00000	40.00000
40.00000	40.00000	40.00000	40.00000	40.00000	40.00000
40.00000	40.00000	40.00000			
	20	8	0.01900		
200.00000	167.54154	137.17704	110.59675	88.81256	72.08412
60.03503	51.88440	46.69865	43.59019	41.83147	40.89040
40.41314	40.18326	40.07784	40.03172	40.01242	40.00468
40.00168	40.00053	40.00000			
	40	9	0.03900		
200.00000	177.23933	155.20179	134.54165	115.78639	99.29813
85.25890	73.67917	64.42534	57.25884	51.87905	47.96330
45.19894	43.30546	42.04644	41.23314	40.72176	40.40711
40.21456	40.09160	40.00000			
	60	9	0.05900		
200.00000	181.46837	163.32715	145.94209	129.63219	114.65246
101.18298	89.32492	79.10333	70.47551	63.34362	57.56936
52.98915	49.42818	46.71205	44.67555	43.16817	42.05653
41.22417	40.56913	40.00000			

III The Error Method

In applying convensional successive numerical scheme for solving parabolic partial differential equations, we may "feel" one feature, i.e., the "passiveness" of the computation scheme. Except selecting the sizes of the independent variables and specifying the initial condition, there is no more handle which can be used to alter or control the development of the computation. Furthermore, the convergent rates of the successive method are often small, especially, to non-linear problems. Frequently, hundreds, or even, thousands iteration loops are required to achieve a single step solution. On account of all these unsatisfactory characteristics with the convensional successive method, the error method is developed. Essentially, the major difference between the successive and the error method is that the later imposes mechanism for estimating and correcting "errors" of trial values of the unknown quantities in a regular successive routing. So the overall computation procedure is greatly shortened. To illustrate the basic scheme of the error method, let us re-consider the unsteady, one-dimensional heat conduction problem of last section. Let $T^{(1)}_{m+1,n}$ represents the first trial temperature profile at m+1 time instant, $T^{*}_{m+1,n}$ represents the "true solution" (still unknown yet). It is clear,

$$T^{*}_{m+1,n} = T^{(1)}_{m+1,n} + \delta_n \qquad (5)$$

where δ_n, the deviation of trial profile, $T^{(1)}_{m+1,n}$, from the "true solution", $T^{*}_{m+1,n}$. To start the iteration process by substituting $T^{(1)}_{m+1,n}$ into the right side of Equation (3), just as same as done in successive method, and decomposing

$T^{(1)}_{m+1,n}$ according to equation (5), we obtain

$$T^{(2)}_{m+1,n} = \frac{T^{*}_{m+1,n+1} + T^{*}_{m+1,n-1}}{2(1+\frac{\Delta X^2}{\Delta t})} + F(T_{m,*}) - \frac{\delta_{n+1}+\delta_{n-1}}{2(1+\frac{\Delta X^2}{\Delta t})} \quad , \quad (6)$$

where $T^{(2)}_{m+1,n}$, the calculated temperature profile at m+1 time instant based on $T^{(1)}_{m+1,n}$. Substracting both sides of equation (6) by $T^{(1)}_{m+1,n}$, we then obtain,

$$R_n = T^{(2)}_{m+1,n} - T^{(1)}_{m+1,n} = \delta_n - \frac{\delta_{n+1}+\delta_{n-1}}{2(1+\frac{\Delta X^2}{\Delta t})} \quad . \quad (7)$$

Equation (7), the error equation, provides N-2 required equations for solving set of N unknowns, δs. The other two equations required are formed from the two boundary conditions. For this particular case, it is obvious that $\delta_1 = \delta_N = 0$. By determining the set of δ values from equation (7), the trial values, $T^{(1)}_{m+1,n}$, can then be modified and and iteration processes repeat, if needed, until the satisfactory "true solution" is achieved. To solve equation (7) and the corresponding boundary conditions for δ, a set of recurrency formular are derived. Let

$$\delta_n = A_n \delta_{n-1} + B_n \quad , \quad (8)$$

where A_n and B_n are undetermined correlation parameters between neighboring δs. Re-arrange equation (7), we obtain,

$$\delta_{n+1} = 2\left(1+\frac{\Delta x^2}{\Delta t}\right)(\delta_n - R_n) - \delta_{n-1} \quad . \quad (9)$$

Eliminating δ_{n-1} from equation (9) and equation (8), we have the recurrency formular,

$$A_{n+1} = 2\left(1+\frac{\Delta x^2}{\Delta t}\right) - \frac{1}{A_n}$$

$$B_{n+1} = \frac{B_n}{A_n} - 2\left(1+\frac{\Delta x^2}{\Delta t}\right) R_n \quad . \qquad (10)$$

To start the operation on the recurrency formular, we know, from equation (9) and boundary condition, $\delta_1 = 0$, that

$$A_3 = 2\left(1+\frac{\Delta x^2}{\Delta t}\right)$$

$$B_3 = -2\left(1+\frac{\Delta x^2}{\Delta t}\right) R_2 \qquad (11)$$

After all the A_n and B_n , for $3 \leq n \leq N$, are determined, since $\delta_N = 0$, therefore,

$$\delta_{N-1} = -\frac{B_N}{A_N} \quad . \qquad (12)$$

The rest δs are calculated from equation (8) in reverse order. The determined δ values are used to modify the trial values of unknown variables, $T^{(1)}_{m+1,n}$, and the iteration cycles repeat, if needed, until satisfactory "true solution" is obtained. It is clear, to this problem, a linear case, equation (7) is a linear equation of δ . However, for non-linear problem, equation (7) would be a non-linear equation of δ . To facilitate the calculation, linearization procedure is used to convert equation (7) into linear form of δ. The other

steps of derivation are identical to those described above.

For clarity, the above description about the derivation of the basic error method will be summarized as follows:

(1) Corresponding to the governing differential equations as well as the boundary conditions, form the implicit finite difference equations, ready to start the regular iteration procedure,

(2) By using the first trial values of the unknown variables, say $T^{(1)}_{m+1,*}$, calculate the unknown variables, say $T^{(2)}_{m+1,*}$,

(3) By recognizing $T^{*}_{m+1,*} = T^{(1)}_{m+1,*} + \delta_{*}$, where $T^{*}_{m+1,*}$, the "true solution", deduce the linear error equations, $R_n = f(\delta_{n+1}, \delta_n, \delta_{n-1})$, where $R_n = T^{(2)}_{m+1,n} - T^{(1)}_{m+1,n}$, from the finite difference numerical scheme,

(4) Let $\delta_{n+1} = A_{n+1}\delta_n + B_{n+1}$, where A_{n+1}, B_{n+1}, the undetermined correlation parameters. Derive the recurrency formular for A_{n+1}, B_{n+1}, i.e.,

$$A_{n+1} = G(A_n)$$

$$B_{n+1} = H(A_n, B_n),$$

from the error equations. Determine the starting A, B values from the related boundary condition,

(5) Calculate A_n, B_n for n up to N. Determine δ_n, first, with the help of one boundary condition and then with the correlation formular,

$$\delta_{n+1} = A_{n+1}\delta_n + B_{n+1},$$

(6) Modify $T^{(1)}_{m+1,*}$ and repeat steps (2) to (5), if needed, until the satisfactory "true solution" is obtained.

With above procedure, a Fortran program, SPD2, was constructed and attached as Appendix B. One case, same as studied by SPD1, was investigated by this program. The results are identical to those from SPD1. Table 2 shows

the comparison of the number of iteration loops and CPU time required by SPD2 and SPD1 for that particular case studied. From Table 2, it appears

Table 2. A comparison of the Number of Iteration Loop and CPU Time Required by Successive and Error Methods for the Problem Studied in Table 1

Program	No. of iteration loop per time step	Total CPU time required for 60 time steps
SPD1	6--9 (most, 9)	5.30 sec
SPD2	1 (occasionally 2, once 3)	4.53 sec

that the CPU time saved by the error method in reference to successive method is only about 20 %, a value not very impressive. However, this observation may also be interpreted as that the successive method is already fairly effective in applying to linear problem.

IV The Application of Error Method to Non-linear Problems

Three non-linear examples are presented to illustrate the application and effectiveness of the error method. To facilitate the presentation, brief qualitative description of these three problems are given first as follows:

(1) A steady laminar tubular flow heat transfer problem with fluid thermal conductivity varying linearly with local temperature. This problem features with non-linear heat conduction term in the transveroe direction to the tube flow.

(2) A steady coaxial laminar tubular flow problem with all the fluid properties varying with local temperature. The problem is sketched in Figure 2. As shown in this diagram, it is clear that radius direction slope of the

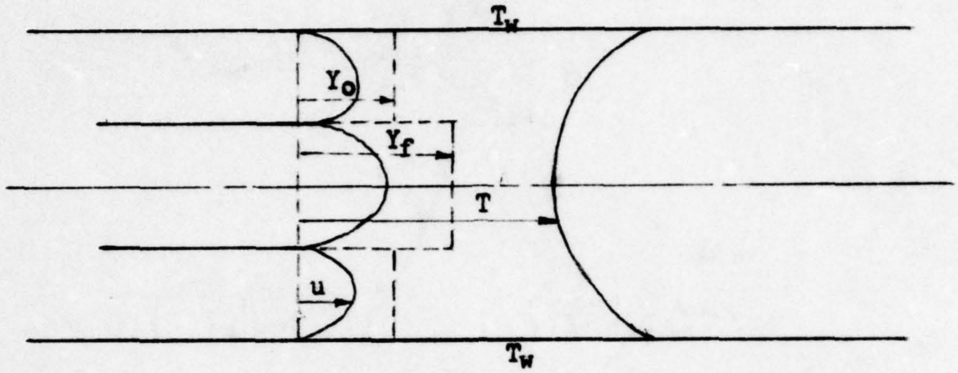
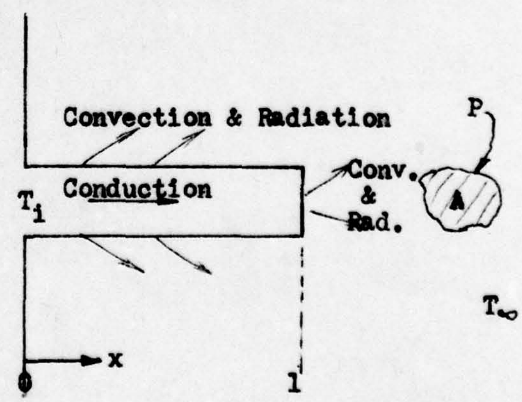


Figure 2. The Coaxial Tubular Flow Sketch

initial velocity profile is discontinuous at inner tube wall location. The concentration profiles of both the inner and outer tube gases are discontinuous at the inner tube wall location. All the difussion terms in the transverse direction are non-linear.

(3) An unsteady, one-dimensional fin problem with radiation. The

Figure 3. The Sketch of a radiative fin



schematic diagram is shown in Figure 3. To this problem, the highly non-linear radiation term presents in both differential equation as well as the boundary condition.

The formulation of all three problems described above are shown

The formulation of all three problems described are shown in Table 3.

Table 3. Formulation of the Three Non-linear Problems

Items	Tube H.T. Problem	Coaxile Tube Problem	Radiative fin Problem
Governing Equations	$\frac{\partial}{\partial x}(\rho u)=0$ $\rho u\frac{\partial T}{\partial x}=\frac{1}{RaPr}\frac{1}{r}\frac{\partial}{\partial r}(Tr\frac{\partial T}{\partial r})$ $\rho T=1$	$\frac{\partial}{\partial x}(\rho u)=0$ $\rho u\frac{\partial T}{\partial x}=\frac{1}{RaPr}\frac{1}{r}\frac{\partial}{\partial r}(Tr\frac{\partial T}{\partial r})$ $\rho u\frac{\partial Y_o}{\partial x}=\frac{1}{ReSc}\frac{1}{r}\frac{\partial}{\partial r}(Tr\frac{\partial Y_o}{\partial r})$ $\rho u\frac{\partial Y_f}{\partial x}=\frac{1}{ReSc}\frac{1}{r}\frac{\partial}{\partial r}(Tr\frac{\partial Y_f}{\partial r})$ $\rho T=1$	$\frac{\partial^2\theta}{\partial x^2}=\frac{\partial\theta}{\partial t}+D\theta$ $\theta\equiv\frac{T-T_\infty}{T_i-T_\infty}$ $D\equiv 1+C_1(\theta^3+4T_r\theta^2+6T_r^2\theta+4T_r^3)$ $C_1\equiv\epsilon\sigma(T_i-T_\infty)^3/h$ $T_r=T_\infty/(T_i-T_\infty)$
Initial and Boundary Conditions	$x=0,\ T=T_0(r)$ $r=0,\ \frac{\partial T}{\partial r}=0$ $r=0.5,\ T=T_w$	$x=0,\ T=T_0(r), Y_o=Y_{o0}(r), Y_f=Y_{f0}(r)$ $r=0,\ \frac{\partial T}{\partial r}=\frac{\partial Y_o}{\partial r}=\frac{\partial Y_f}{\partial r}=0$ $r=0.5,\ T=T_w, \frac{\partial Y_o}{\partial r}=\frac{\partial Y_f}{\partial r}=0.$	$t=0,\ \theta=1$ $x=0,\ \theta=1$ $x=ml,\ \frac{\partial\theta}{\partial x}=-C_cD\theta$ $ml\equiv l\sqrt{\frac{hP}{kA}}$, dimensionless fin length $C_c\equiv\frac{h}{km}$, Biot number
Finite Difference equations	$T_{2,j}=\{(\frac{C_pC_j}{4})^2+C_pC_jT_{1,j}$ $+\frac{J_{12}}{2}(T_{1,j+1}+T_{2,j+1})^2$ $+\frac{J_{32}}{2}(T_{1,j-1}+T_{2,j-1})^2\}^{\frac{1}{2}}$ $-T_{1,j}-\frac{C_pC_j}{4}$, $C_j\equiv(\rho u)_j$ $C_p\equiv\frac{8\Delta r^2 RaPr}{\Delta x}$ $J_{12}=(J-\frac{1}{2})/(J-1)$ $J_{32}=(J-\frac{3}{2})/(J-1)$ $T_{2,N}=T_w$	Temperature expressions are identical to those in left column, $Y_{2,j}=\frac{J_{12}T_{+1}}{C_{sj}}(Y_{1,j+1}+Y_{2,j+1})$ $+\frac{J_{32}T_{-1}}{C_{sj}}(Y_{1,j-1}+Y_{2,j-1})$ $+\frac{C_sC_j-J_{12}T_{+1}-J_{32}T_{-1}}{C_{sj}}Y_{1,j}$ $C_{sj}=C_sC_j+J_{12}T_{+1}+J_{32}T_{-1}$ $C_s=\frac{8\Delta r^2 ReSc}{\Delta x}$ $T_{+1}=T_{1,j+1}+T_{2,j+1}+T_{1,j-1}+T_{2,j-1}$ $T_{-1}=T_{1,j}+T_{2,j}+T_{1,j-1}+T_{2,j-1}$ $J_{12}=(J-\frac{1}{2})/(J-1)$ $J_{32}=(J-\frac{3}{2})/(J-1)$ Y can be Y_o or Y_f	$\theta_{2,j}=\frac{\theta_{\pm1}}{2\Delta x^2C_d}-\frac{C_u}{C_d}\theta_{1,j}$ $\theta_{\pm1}=\theta_{2,j+1}+\theta_{2,j-1}+\theta_{1,j+1}+\theta_{1,j-1}$ $C_d=1/\Delta x^2+1/\Delta t+0.5D_{1\frac{1}{2},j}$ $C_u=1/\Delta x^2-1/\Delta t+0.5D_{1\frac{1}{2},j}$ $\theta_{2,N}=\frac{1}{C_e}\{\frac{2\theta_{1\frac{1}{2},N-1}-\theta_{1,N}}{2\Delta x}-\frac{\Delta x}{2\Delta t}(\frac{\theta_{1,N-1}}{4}-\theta_{1,N'})$ $-C_c[\frac{D_N\theta_{1,N}}{2}+D_{N'}\frac{\Delta x}{b}(\frac{\theta_{1\frac{1}{2},N-1}}{4}+\frac{3\theta_{1,N}}{8})]\}$ $C_e\equiv\frac{1}{2\Delta x}+\frac{3\Delta x}{8\Delta t}+C_c(\frac{D_N}{2}+\frac{3D_{N'}}{8}\cdot\frac{\Delta x}{b})$ N', the location $\frac{1}{4}\Delta x$ from N point

(Continue at next page)

Table 3. (continued)

Items	Tube H.T.Problem	Coaxile Tube Problem	Radiative Fin Problem
Linearized Error Equations	$\delta_{j+1} \simeq C_a(\delta_j - R_j) - C_b \delta_{j-1}$ $C_a \equiv \frac{2\,C_{au}}{J_{12}(T_{1,j+1}+T_{2,j+1})}$ $C_{au} = \left\{ \left(\frac{C_p C_j}{4}\right)^2 + C_p C_j T_{1,j} + \frac{J_{12}}{2}(T_{1,j+1}+T_{2,j+1})^2 + \frac{J_{32}}{2}(T_{1,j-1}+T_{2,j-1})^2 \right\}^{\frac{1}{2}}$ $C_b \equiv \frac{J_{32}(T_{1,j-1}+T_{2,j-1})}{J_{12}(T_{1,j+1}+T_{2,j+1})}$ $R_j \equiv T_{2,j}^{(2)} - T_{2,j}^{(1)}$	Temperature error eqs. are identical to those in left column, $\delta_{j+1} = C_a(\delta_j - R_j) - C_b \delta_{j-1}$ $C_a = \frac{C_{sj}}{J_{12} \cdot T_{+1}}$ $C_b = \frac{J_{32} \cdot T_{-1}}{J_{12} \cdot T_{+1}}$ $R_j = Y_{2,j}^{(2)} - Y_{2,j}^{(1)}$ δ can be δ_f or δ_o R can be R_f or R_o.	$\delta_{j+1} \simeq C_a \delta_j - \delta_{j-1} - C_b R_j$ $C_a = 2\Delta x^2 C_d + \frac{\Delta t \theta_{t1} + 4\Delta x^2}{2\Delta t C_d} C_1 \left[\frac{3}{2}\theta_{1\frac{1}{2},j}^2 + 4T_r \theta_{1\frac{1}{2},j} + 3T_r^2\right]$ $C_b = 2\Delta x^2 C_d$ $R_j = \theta_{2,j}^{(2)} - \theta_{2,j}^{(1)}$ $\delta_N = S_g \delta_{N-1} + R_N C_e / S_{gd}$ $S_g \equiv S_{gu}/S_{gd}$ $S_{gu} = 1/2\Delta x - \Delta x/8\Delta t - 0.25\, C_c \frac{\Delta x}{b} \cdot \left[\frac{3\theta_{1,N} C_{D(N-1)}}{8} + \frac{D_{N'}}{2} + \frac{\theta_{1\frac{1}{2},N-1} \cdot C_{D(N-1)}}{4} + \frac{3\theta_{2,N} C_{D(N-1)}}{8}\right]$ $S_{gd} = \frac{1}{2\Delta x} + \frac{3\Delta x}{8\Delta t} + C_c \left\{ \frac{D_N}{2} + \frac{3\Delta x D_{N'}}{8b} + \left[\left(\frac{1}{2} + \frac{9}{32}\frac{\Delta x}{b}\right)\theta_{2,N} + \frac{\theta_{1,N}}{2} + \frac{9}{32}\frac{\Delta x}{b}\theta_{1,N} + \frac{3\Delta x}{16b} \cdot \theta_{1\frac{1}{2},N-1}\right] C_{DN} \right\}$ $C_{Dj} = C_1\left(\frac{3}{2}\theta_{1\frac{1}{2},j}^2 + 4T_r \theta_{1\frac{1}{2},j} + 3T_r^2\right)$
Recurrency formular	$A_{j+1} = C_a - \frac{C_b}{A_j}$ $B_{j+1} = \frac{C_b B_j}{A_j} - C_a R_j$	$A_{j+1} = C_a - \frac{C_b}{A_j}$ $B_{j+1} = \frac{C_b B_j}{A_j} - C_a R_j$ A, B, R, C_a, C_b are for all T, Y_f and Y_o fields.	$A_{j+1} = C_a - \frac{1}{A_j}$ $B_{j+1} = \frac{B_j}{A_j} - C_a R_j$
Information derived from boundary conditions	$A_2 = 1.0$, $B_2 = 0.0$ $\delta_{N-1} = -\frac{B_N}{A_N}$	For temperature : $A_2 = 1.0$, $B_2 = 0.0$ $\delta_{N-1} = -\frac{B_N}{A_N}$ For Y_f and Y_o : $A_2 = 1.0$, $B_2 = 0.0$ $\delta_N = \delta_{N-1} = \frac{B_N}{1 - A_N}$ all A, B, δ can be applied to Y_f and Y_o fields.	$A_2 = 1.0$, $B_2 = 0.0$ $\delta_{N-1} = \frac{B_N - \frac{R_N C_e}{S_{gd}}}{S_g - A_N}$

(continued on next page)

Table 3. (continued)

Items	Tube H.T. Problem	Coaxile Tube Problem	Radiative Fin Problem
Modified Trial Profile for next iteration, if needed	$T_{2j}^{(1)'} \equiv T_{2j}^{(1)} + \delta_j$	$T_{2j}^{(1)'} = T_{2j}^{(1)} + \delta_{jT}$ $Y_{f2j}^{(1)'} = Y_{f2j}^{(1)} + \delta_{jY_f}$ $Y_{o2j}^{(1)'} = Y_{o2j}^{(1)} + \delta_{jY_o}$	$\Theta_{2j}^{(1)'} = \Theta_{2j}^{(1)} + \delta_j$
Nomenclature unspec. by above and left expressions,	ρ, fluid density u, fluid velocity T, fluid temperature x, tube axile coor. r, tube rad coor. Re, Reynolds number Pr, Prandtl number $T_o(r)$, entrance temperature Profile T_w, tube wall temperature j (or J), jth node point along r-direction N, total number of nodal points along r-direction δ, deviation of trial value from "true solution."	Y_o, outer tube gas conc. Y_f, inner tube gas conc. $Y_{oo}(r)$, entrance Y_o profile $Y_{fo}(r)$, entrance Y_f Profile Sc, Schmidt number	T_1, initial and fin root temperature T_∞, environment temp. ϵ, emmisivity σ, Stef-Boltzmann Cons. h, conv. h. t. Coef. l, fin length P, fin cross section perimeter length k, fin thermal cond. A, fin cross section area

Based upon formular in Table 3, computer programs with error method were designed for the above three problems as SPD6,SPD4 and SPD8. They are attached as Appendix F,D and H. Three corresponding programs (SPD5,SPD3 and SPD7),based on successive method, for these three problems are also constructed and attached as Appendix E,C and G. Computational effectiveness of these programs were investigated on several sample cases. Table 4 shows several computational effectiveness of various programs run.

Table 4 Comparison of CPU time and Number of Iteration Loop required by Successive and Error Method for the three non-linear problems

Items		Tube Flow	Coaxile Flow	Radiative Fin
Given Conditions		Re=500.00 Pr=1.0 T_w=3.0 N=41 M (total steps calculated)= 80	Re=500.00 Pr=1.0 Sc=1.0 T_w=3.0 N=41 M=40	ml=1.0 C =2.0 C =0.2 T_r=0.4 B(fin th.)=0.05 N=41 M=80
Program used	Succ	SPD5	SPD3	SPD7
	Error	SPD6	SPD4	SPD8
No. of Loops used per step	Succ	49-73	85-127	7-9
	Error	3	4	4
Total CPU Time	Succ	53.85 sec	67.70 sec	29.05 sec
	Error	5.55 sec*	30.18 sec	4.71 sec*

*print only 1/10th as many lines as did by the comparable programs.

Table 4 indicates that, by using error method, the CPU time used can be reduced to about half or less in reference to those corresponding calculation done by successive method.

V Discussion and Conclusion

(1) Investigation in previous sections has indicated that the error method developed up to this stage does provide us an alternative way for solving parabolic partial differential equations. The utilisation of this method, in particular to non-linear problems, results in significant reducing CPU time and number of iteration loops required. This feature , together with its wide applicability may turn this method into an attractive tool.

(2) The recurrency technique suggested here can be considered as an effective method for solving the three line matrix .

(3) The applicability of the error method to ordinary differential equations, linear or non-linear, is obvious.

(4) To linear parabolic type equation, the above recurrency technique can be applied directly to the unknown parameters instead of the errors of the trial values of the unknown parameters. It thus provides another alternative method. To non-linear equation, however, the direct application of the recurrency technique to the unknown parameters may not be able to lead to numerical scheme which can convergence as fast as the current error method.

(5) Along the line of designing new computational method, a systematic exploitation of the applicability of the error method to problems with elliptical type partial differential equation may not be inappropriate.

APPENDIX A

```
DAY
Tuesday, November 13, 1979 16:31:02
@TY SPD1.FOR
C        SPD1--SOLUTION FOR LINEAR HEAT TRANSFER PROBLEM,
C        ONE DIMENSIONAL UNSTEADY HEAT CONDUCTION
C          SUCCESSIVE METHOD              SEPTEMBER 17,1979
      DIMENSION U(101),UN(101),RS(101),UNO(101)
        OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
      READ(1,10) N,MAX
   10 FORMAT(4I10)
      READ(1,20) U(1),UI
   20 FORMAT(8F10.5)
        DY=1.0/FLOAT(N-1)
        DT=DY*DY/2.5
        DO 30 I=2,N
        U(I)=UI
   30   CONTINUE
      M=1
        TIME=0.0
      WRITE(5,150) M,NL,TIME,(U(I),I=1,N)
      DO 40 I=1,N
      UN(I)=U(I)
      UNO(I)=UN(I)
   40 CONTINUE
      N1=N-1
      CF1=2.0*(1.0+DY*DY/DT)
      CF2=1.0+DT/(DY*DY)
   50 NL=0
      DO 60 I=2,N1
      IP=I+1
      I1=I-1
      RS(I)=(U(IP)+U(I1)-2.0*U(I))/CF1+U(I)/CF2
   60 CONTINUE
   70 NT=0
      DO 90 I=2,N1
      IP=I+1
      I1=I-1
      UN(I)=(UN(IP)+UN(I1))/CF1+RS(I)
      DNO=ABS(UN(I)-UNO(I))
      IF(DNO-0.000001) 85,85,80
   80 NT=1
   85 UNO(I)=UN(I)
   90 CONTINUE
      IF(NT-1) 120,100,100
  100 NL=NL+1
      GO TO 70
  120 M=M+1
        TIME=TIME+DT
      DO 130 I=1,N
      U(I)=UN(I)
      UNO(I)=UN(I)
  130 CONTINUE
      WRITE(5,150) M,NL,TIME,(UN(I),I=1,N)
  150 FORMAT(10X,2I10,F10.5,/20(5X,6(F9.5,1X)/))
      IF(M-MAX) 50,200,200
  200 STOP
      END

@
```

APPENDIX B

```
DAY
Wednesday, November 14, 1979 11:37:07
@TY SPD2.FOR
C          SPD2--NEW METHOD FOR LINEAR HEAT TRANSFERPROBLEM,
C          ONF DIMENSIONAL UNSTEADY HEAT CONDUCTION
C            ERROR METHOD
         DIMENSION U(101),UN(101),RS(101),A(101),B(101),
        1R(101),DEL(101)
           OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
         READ(1,10) N,MAX
      10 FORMAT(4I10)
           READ (1,20) U(1),UI
      20 FORMAT(8F10.5)
           DY=1.0/FLOAT(N-1)
           DT=DY*DY/2.5
           DO 30 I=2,N
           U(I)=UI
      30   CONTINUE
         M=1
           TIME=0.0
         WRITE(5,150) M,NL,TIME,(U(I),I=1,N)
         DO 40 I=1,N
         UN(I)=U(I)
      40 CONTINUE
         N1=N-1
         CF1=2.0*(1.0+DY*DY/DT)
         CF2=1.0+DT/(DY*DY)
      50 NL=0
         DO 60 I=2,N1
         IP=I+1
         I1=I-1
         RS(I)=(U(IP)+U(I1)-2.0*U(I))/CF1+U(I)/CF2
      60 CONTINUE
      70 NT=0
         A(2)=10.0**25
         B(2)=0.0
         DEL(N)=0.0
         DO 130 I=2,N1
         IP=I+1
         I1=I-1
         UNI=(UN(IP)+UN(I1))/CF1+RS(I)
         DNO=ABS(UNI-UN(I))
         R(I)=UNI-UN(I)
         A(I+1)=CF1-1.0/A(I)
      90 B(I+1)=B(I)/A(I)-CF1*R(I)
      93   IF(DNO.LE.0.00001) GO TO 130
         NT=1
     130 CONTINUE
         IF(NT-1)200,140,140
     140 DO 180 I=2,N1
         IR1=N-I+1
         IR2=IR1+1
         DEL(IR1)=DEL(IR2)/A(IR2)-B(IR2)/A(IR2)
         UN(IR1)=UN(IR1)+DEL(IR1)
     180 CONTINUE
         NL=NL+1
         GO TO 70
     200 M=M+1
           TIME=TIME+DT
         DO 230 I=1,N
         U(I)=UN(I)
     230 CONTINUE
         WRITE(5,150) M,NL,TIME,(UN(I),I=1,N)
```

```
  150 FORMAT(10X,2I10;F10.5;/20(5X;6(F9.5;1X)/))
      IF(M-MAX) 50,300;300
  300 STOP
      END
@
```

APPENDIX C

```
DAY
Tuesday, November 13, 1979 19:09:36
@TY SPD3.FOR
C      SPD3--DIFFUSION IN DOUBLE TUBE FLOW PROBLEM,JULY/31/79
C        SUCCESSIVE METHOD
      DIMENSION T(2,101),YF(2,101),YO(2,101),U(2,101),RO(2,101),
     1C(101),DLT(101),DLTN(101),ROT(101),TN(101),YFN(101),
     2YON(101),ROYF(101),ROYO(101),DLYF(101),DYFN(101),
     3R(101),DLYO(101),DYON(101)
      OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
      READ(1,10) M,N,RE,PR,SC,TW,WP,CJ
   10 FORMAT(2I10,10F10.5)
      DR=0.5/FLOAT(N-1)
        DX=DR*DR*RE/2.5
      WRITE(5,15) M,N,DX,RE,PR,SC,TW,WP,CJ
   15 FORMAT(2I5,8(F10.5,1X)/)
C     BOUNDARY CONDITIONS
      R(1)=0.0
      N1=N-1
      DO 16 J=1,N1
      J1=J+1
      R(J1)=R(J)+DR
   16 CONTINUE
      LNI=0.5*WP/DR+1.0
      COP=8.0*DR**2*RE*PR/DX
      COS=COP*SC/PR
      DO 30 J=1,N
      AJ=J-1
      DO 25 I=1,2
      T(I,J)=1.0+(TW-1.0)*(AJ*DR*2.0)**2
      RO(I,J)=1.0/T(I,J)
      IF(J-LNI) 17,17,19
   17 YF(I,J)=1.0
      YO(I,J)=0.0
      U(I,J)=CJ*(1.0-(2.0*R(J)/WP)**2)
      GO TO 25
   19 YF(I,J)=0.0
      YO(I,J)=0.5
      U(I,J)=1.0-(2.0*R(J))**2+(1.0-WP**2)*ALOG(1.0/(2.0*R(J)
     1))/ALOG(WP)
   25 CONTINUE
      C(J)=U(1,J)*RO(1,J)
   30 CONTINUE
      C(N)=0.0
      C(2)=C(1)
      C(LNI)=0.25*(C(LNI+1)+C(LNI-1))
      U(1,N)=0.0
      U(2,N)=0.0
      YO(1,LNI)=0.25
      YO(2,LNI)=0.25
      YF(1,LNI)=0.5
      YF(2,LNI)=0.5
      IK=1
      SMTUR=0.0
      SMUR=0.0
      DO 33 J=2,N1
      AJ1=J-1
      SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
      SMUR=SMUR+U(1,J)*AJ1
   33 CONTINUE
      TBBR=SMTUR/SMUR
      QW=260.0*(TW-T(1,N1))/DR
      ANULT=QW/(260.0*(TW-TBBB))
```

```
      WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
      WRITE(5,570) ANULT,QW
      WRITE(5,950) (U(1,J),J=1,N)
      WRITE(5,565) X,(RO(1,J),J=1,N)
      WRITE(5,566) (YF(1,J),J=1,N)
      WRITE(5,568) (YO(1,J),J=1,N)
      X=0.0
   43 IK=IK+1
      IF(IK-M)46,46,1703
   46 X=X+DX
C     UNKNOWS NEAR THE CENTRAL LINE(CENTRAL B.C.)
      NIT=1
   48 DO 57 I=1,2
      T(I,2)=T(I,1)
      YF(I,2)=YF(I,1)
      YO(I,2)=YO(I,1)
   57 CONTINUE
C     UNKNOWNS IN THE FIELD(2,N1)
   61 NT=0
      DO 1300 J=2,N1
      J1=J-1
      AJ1=J1
      AJ12=(AJ1+0.5)/AJ1
      AJ32=(AJ1-0.5)/AJ1
      CCOP=COP*C(J)+4.0*T(1,J)
      SQT2=SQRT((0.25*COP*C(J))**2+COP*C(J)*T(1,J)+0.5*AJ12*
     1(T(1,J+1)+T(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
      TN(J)=-0.25*CCOP+SQT2
        TUS=T(1,J+1)+T(2,J+1)+T(1,J)+T(2,J)
        TLS=T(1,J-1)+T(2,J-1)+T(1,J)+T(2,J)
        CO2J=COS*C(J)+AJ12*TUS+AJ32*TLS
        YFN(J)=(AJ12*TUS*(YF(1,J+1)+YF(2,J+1))+AJ32*TLS*
     1  (YF(1,J-1)+YF(2,J-1))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
     2  YF(1,J))/CO2J
        YON(J)=(AJ12*TUS*(YO(1,J+1)+YO(2,J+1))+AJ32*TLS*
     1  (YO(1,J-1)+YO(2,J-1))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
     2  YO(1,J))/CO2J
      ROT(J)=TN(J)-T(2,J)
      ROYF(J)=YFN(J)-YF(2,J)
      ROYO(J)=YON(J)-YO(2,J)
      AROT=ABS(ROT(J))
      AROYF=ABS(ROYF(J))
      AROYO=ABS(ROYO(J))
      IF(AROT-0.000001) 572,572,1150
  572 IF(AROYF-0.000001)590,590,1150
  590 IF(AROYO-0.000001)1300,1300,1150
 1150 NT=1
 1300 CONTINUE
      IF(NT) 1567,1567,1330
 1330 NIT=NIT+1
      DO 1538 J=2,N1
        T(2,J)=TN(J)
        YF(2,J)=YFN(J)
        YO(2,J)=YON(J)
 1538 CONTINUE
      T(2,N)=TW
      YF(2,N)=YF(2,N1)
      YO(2,N)=YO(2,N1)
        T(2,1)=T(2,2)
        YF(2,1)=YF(2,2)
        YO(2,1)=YO(2,2)
      GO TO 61
 1567 QW=260.0*(TW-T(2,N1))/DR
        DO 919 J=1,N
        RO(2,J)=1.0/T(2,J)
        U(2,J)=C(J)/RO(2,J)
```

```
  919   CONTINUE
        SMTUR=0.0
        SMUR=0.0
        DO 908 J=2,N1
        J1=J-1
        SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
        SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
  908 CONTINUE
        TBBB=SMTUR/SMUR
        ANULT=QW/(260.0*(TW-TBBB))
        WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
        WRITE(5,570) ANULT,QW
        WRITE(5,950) (U(2,J),J=1,N)
        WRITE(5,565) X,(RO(2,J),J=1,N)
        WRITE(5,566) (YF(2,J),J=1,N)
        WRITE(5,568)(YO(2,J),J=1,N)
  535 FORMAT(2X,' T,QW',2I5,5(F9.5,1X)/25
     1  (17X,5(F9.5,1X)/))
  570 FORMAT(17X,F9.5,1X,F9.3)
  950 FORMAT(2X,'     U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
  565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
  566 FORMAT(2X,'    YF',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
  568 FORMAT(2X,'    YO',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
        DO 1679 J=1,N
        T(1,J)=T(2,J)
        YF(1,J)=YF(2,J)
        YO(1,J)=YO(2,J)
        RO(1,J)=1.0/T(1,J)
        U(1,J)=C(J)/RO(1,J)
 1679 CONTINUE
 1697 GO TO 43
 1703 STOP
        END
@
```

APPENDIX D

```
DAY
Tuesday, November 13, 1979 19:08:00
@TY SPD4.FOR
C       SPD4--DIFFUSION IN DOUBLE TUBE FLOW PROBLEM,JULY/31/79
C         ERROR METHOD
        DIMENSION T(2,101),YF(2,101),YO(2,101),U(2,101),RO(2,101),
       1C(101),TN(101),YFN(101),
       2YON(101),AT(101),BT(101),AY(101),BYF(101),
       3R(101),BYO(101),DEL(101),DEF(101),DEO(101)
        OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
        READ(1,10) M,N,RE,PR,SC,TW,WP,CJ
     10 FORMAT(2I10,10F10.5)
        DR=0.5/FLOAT(N-1)
          DX=DR*DR*RE/2.5
        WRITE(5,15) M,N,DX,RE,PR,SC,TW,WP,CJ
     15 FORMAT(2I5,8(F10.5,1X)/)
C       BOUNDARY CONDITIONS
        R(1)=0.0
        N1=N-1
        DO 16 J=1,N1
        J1=J+1
        R(J1)=R(J)+DR
     16 CONTINUE
        LNI=0.5*WP/DR+1.0
        COP=8.0*DR**2*RE*PR/DX
        COS=COP*SC/PR
        DO 30 J=1,N
        AJ=J-1
        DO 25 I=1,2
        T(I,J)=1.0+(TW-1.0)*(AJ*DR*2.0)**2
        RO(I,J)=1.0/T(I,J)
        IF(J-LNI) 17,17,19
     17 YF(I,J)=1.0
        YO(I,J)=0.0
        U(I,J)=CJ*(1.0-(2.0*R(J)/WP)**2)
        GO TO 25
     19 YF(I,J)=0.0
        YO(I,J)=0.5
        U(I,J)=1.0-(2.0*R(J))**2+(1.0-WP**2)*ALOG(1.0/(2.0*R(J)
       1))/ALOG(WP)
     25 CONTINUE
        C(J)=U(1,J)*RO(1,J)
     30 CONTINUE
        C(N)=0.0
          C(2)=C(1)
          C(LNI)=0.25*(C(LNI+1)+C(LNI-1))
        U(1,N)=0.0
        U(2,N)=0.0
        YO(1,LNI)=0.25
        YO(2,LNI)=0.25
        YF(1,LNI)=0.5
        YF(2,LNI)=0.5
        IK=1
        SMTUR=0.0
        SMUR=0.0
        DO 33 J=2,N1
        AJ1=J-1
        SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
        SMUR=SMUR+U(1,J)*AJ1
     33 CONTINUE
        TBBB=SMTUR/SMUR
        QW=260.0*(TW-T(1,N1))/DR
        ANULT=QW/(260.0*(TW-TBBB))
```

```
      WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
      WRITE(5,570) ANULT,QW
      WRITE(5,950) (U(1,J),J=1,N)
      WRITE(5,565) X,(RO(1,J),J=1,N)
      WRITE(5,566) (YF(1,J),J=1,N)
      WRITE(5,568) (YO(1,J),J=1,N)
      X=0.0
   43 IK=IK+1
      IF(IK-M)46,46,1703
   46 X=X+DX
C     UNKNOWS NEAR THE CENTRAL LINE(CENTRAL B.C.)
      NIT=1
   48 DO 57 I=1,2
      T(I,2)=T(I,1)
      YF(I,2)=YF(I,1)
      YO(I,2)=YO(I,1)
   57 CONTINUE
C     UNKNOWNS IN THE FIELD(2,N1)
   61 NT=0
      AT(2)=1.0
      BT(2)=0.0
      DO 1300 J=2,N1
      J1=J-1
      AJ1=J1
      AJ12=(AJ1+0.5)/AJ1
      AJ32=(AJ1-0.5)/AJ1
      CCOP=COP*C(J)+4.0*T(1,J)
      SQT2=SQRT((0.25*COP*C(J))**2+COP*C(J)*T(1,J)+0.5*AJ12*
     1(T(1,J+1)+T(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
      TN(J)=-0.25*CCOP+SQT2
      ROT=TN(J)-T(2,J)
      AROT=ABS(ROT)
      CAT=2.0*SQT2/(AJ12*(T(1,J+1)+T(2,J+1)))
      CBT=AJ32*(T(1,J-1)+T(2,J-1))/(AJ12*(T(1,J+1)+T(2,J+1)))
      AT(J+1)=CAT-CBT/AT(J)
      BT(J+1)=CBT*BT(J)/AT(J)-CAT*ROT
  345 IF(AROT-0.000001) 1300,1300,1150
 1150 NT=1
 1300 CONTINUE
      IF(NT.LE.0) GO TO 1567
      DEL(N1)=-BT(N)/AT(N)
      T(2,N1)=T(2,N1)+DEL(N1)
      DO 1313 J=2,N1
      NJ=N-J
      NJ1=N+1-J
      DEL(NJ)=(DEL(NJ1)-BT(NJ1))/AT(NJ1)
      T(2,NJ)=T(2,NJ)+DEL(NJ)
 1313 CONTINUE
      T(2,1)=T(2,2)
      T(2,N)=TW
      NIT=NIT+1
      GO TO 61
 1567 NY=0
        AY(2)=1.0
        BYF(2)=0.0
        BYO(2)=0.0
        DO 1400 J=2,N1
        J1=J-1
        AJ1=J1
        AJ12=(AJ1+0.5)/AJ1
        AJ32=(AJ1-0.5)/AJ1
        TUS=T(1,J+1)+T(2,J+1)+T(1,J)+T(2,J)
        TLS=T(1,J)+T(2,J)+T(1,J-1)+T(2,J-1)
        CO2J=COS*C(J)+AJ12*TUS+AJ32*TLS
        YFN(J)=(AJ12*TUS*(YF(1,J+1)+YF(2,J+1))+AJ32*TLS*
     1  (YF(1,J-1)+YF(2,J-1))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
```

```
     2 YF(1,J))/CO2J
       YON(J)=(AJ12*TUS*(YO(1,J+1)+YO(2,J+1))+AJ32*TLS*
     1 (YO(1,J-1)+YO(2,J-1))+(COS*C(J)-AJ12*TUS-AJ32*TLS)*
     2 YO(1,J))/CO2J
       ROYF=YFN(J)-YF(2,J)
       ROYO=YON(J)-YO(2,J)
       AROYF=ABS(ROYF)
       AROYO=ABS(ROYO)
       CAY=CO2J/(AJ12*TUS)
       CBY=AJ32*TLS/(AJ12*TUS)
       AY(J+1)=CAY-CBY/AY(J)
       BYF(J+1)=CBY*BYF(J)/AY(J)-CAY*ROYF
       BYO(J+1)=CBY*BYO(J)/AY(J)-CAY*ROYO
       IF(AROYF.LE.0.000001) GO TO 1395
       NY=1
  1395 IF(AROYO.LE.0.000001) GO TO 1400
       NY=1
  1400 CONTINUE
       IF(NY.LE.0) GO TO 1576
       DEF(N1)=BYF(N)/(1.0-AY(N))
       DEO(N1)=BYO(N)/(1.0-AY(N))
       YF(2,N1)=YF(2,N1)+DEF(N1)
       YO(2,N1)=YO(2,N1)+DEO(N1)
       DO 1414 J=2,N1
       NJ=N-J
       NJ1=N+1-J
       DEF(NJ)=(DEF(NJ1)-BYF(NJ1))/AY(NJ1)
       DEO(NJ)=(DEO(NJ1)-BYO(NJ1))/AY(NJ1)
       YF(2,NJ)=YF(2,NJ)+DEF(NJ)
       YO(2,NJ)=YO(2,NJ)+DEO(NJ)
  1414 CONTINUE
       YF(2,1)=YF(2,2)
       YF(2,N)=YF(2,N1)
       YO(2,1)=YO(2,2)
       YO(2,N)=YO(2,N1)
       NIT=NIT+1
       GO TO 1567
  1576 QW=260.0*(TW-T(2,N1))/DR
       DO 919 J=1,N
       RO(2,J)=1.0/T(2,J)
       U(2,J)=C(J)/RO(2,J)
   919 CONTINUE
       SMTUR=0.0
       SMUR=0.0
       DO 908 J=2,N1
       J1=J-1
       SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
       SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
   908 CONTINUE
       TBBB=SMTUR/SMUR
       ANULT=QW/(260.0*(TW-TBBB))
       WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
       WRITE(5,570) ANULT,QW
       WRITE(5,950) (U(2,J),J=1,N)
       WRITE(5,565) X,(RO(2,J),J=1,N)
       WRITE(5,566) (YF(2,J),J=1,N)
       WRITE(5,568)(YO(2,J),J=1,N)
   535 FORMAT(2X,' T,QW',2I5,5(F9.5,1X)/25
     1 (17X,5(F9.5,1X)/))
   570 FORMAT(17X,F9.5,1X,F9.3)
   950 FORMAT(2X,'     U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
   565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
   566 FORMAT(2X,'    YF',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
   568 FORMAT(2X,'    YO',10X,5(F9.5,1X)/25(17X,5(F9.5,1X)/))
       DO 1679 J=1,N
       T(1,J)=T(2,J)
```

```
      YF(1,J)=YF(2,J)
      YO(1,J)=YO(2,J)
      RO(1,J)=1.0/T(1,J)
      U(1,J)=C(J)/RO(1,J)
 1679 CONTINUE
 1697 GO TO 43
 1703 STOP
      END
@
```

APPENDIX E

DAY
Tuesday, November 13, 1979 20:03:13
@TY SPD5.FOR

```
      C         SPD5--NON LINEAR TUBE FLOW HEAT TRANSFER PROBLEM,
      C            SUCCESSIVE METHOD            NOVEMBER 1,1979
            DIMENSION T(2,101),U(2,101),RO(2,101),
           1C(101),TN(101),
           3R(101)
            OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
            READ(1,10) M,N,RE,PR,TW
         10 FORMAT(2I10,10F10.5)
            DR=0.5/FLOAT(N-1)
              DX=DR*DR*RE/2.5
            WRITE(5,15) M,N,DX,RE,PR,TW
         15 FORMAT(2I5,8(F10.5,1X)/)
      C     BOUNDARY CONDITIONS
            R(1)=0.0
            N1=N-1
            DO 16 J=1,N1
            J1=J+1
            R(J1)=R(J)+DR
         16 CONTINUE
            COP=8.0*DR**2*RE*PR/DX
            DO 30 J=1,N
            AJ=J-1
            DO 25 I=1,2
            T(I,J)=1.0+(TW-1.0)*(AJ*DR*2.0)**2
            RO(I,J)=1.0/T(I,J)
            U(I,J)=1.0-(2.0*R(J))**2
         25 CONTINUE
            C(J)=U(1,J)*RO(1,J)
         30 CONTINUE
            C(N)=0.0
            U(1,N)=0.0
            U(2,N)=0.0
            IK=1
            SMTUR=0.0
            SMUR=0.0
            DO 33 J=2,N1
            AJ1=J-1
            SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
            SMUR=SMUR+U(1,J)*AJ1
         33 CONTINUE
            TBBB=SMTUR/SMUR
            QW=260.0*(TW-T(1,N1))/DR
            ANULT=QW/(260.0*(TW-TBBB))
            WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
            WRITE(5,570) ANULT,QW
            WRITE(5,950) (U(1,J),J=1,N)
            WRITE(5,565) X,(RO(1,J),J=1,N)
            X=0.0
         43 IK=IK+1
            IF(IK-M)46,46,1703
         46 X=X+DX
      C     UNKNOWS NEAR THE CENTRAL LINE(CENTRAL B.C.)
            NIT=1
         48 DO 57 I=1,2
            T(I,2)=T(I,1)
            U(I,2)=U(I,1)
            RO(I,2)=RO(I,1)
         57 CONTINUE
      C     UNKNOWNS IN THE FIELD(2,N1)
         61 NT=0
```

```
      DO 1300 J=2,N1
      J1=J-1
      AJ1=J1
      AJ12=(AJ1+0.5)/AJ1
      AJ32=(AJ1-0.5)/AJ1
      CCOP=COP*C(J)+4.0*T(1,J)
      SQT2=SQRT((0.25*COP*C(J))**2+COP*C(J)*T(1,J)+0.5*AJ12*
     1(T(1,J+1)+T(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
      TN(J)=-0.25*CCOP+SQT2
      ROT=TN(J)-T(2,J)
      AROT=ABS(ROT)
      IF(AROT-0.000001) 1300,1300,1150
 1150 NT=1
 1300 CONTINUE
      IF(NT.LE.0) GO TO 1567
      DO 1313 J=2,N1
      T(2,J)=TN(J)
 1313 CONTINUE
      T(2,1)=T(2,2)
      NIT=NIT+1
      GO TO 61
 1567 QW=260.0*(TW-T(2,N1))/DR
        DO 919 J=1,N
        RO(2,J)=1.0/T(2,J)
        U(2,J)=C(J)/RO(2,J)
  919   CONTINUE
      SMTUR=0.0
      SMUR=0.0
      DO 908 J=2,N1
      J1=J-1
      SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
      SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
  908 CONTINUE
      TBBB=SMTUR/SMUR
      ANULT=QW/(260.0*(TW-TBBB))
      WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
      WRITE(5,570) ANULT,QW
      WRITE(5,950) (U(2,J),J=1,N)
      WRITE(5,565) X,(RO(2,J),J=1,N)
  535 FORMAT(2X,' T,QW',2I5,5(F9.5,1X)/25
     1  (17X,5(F9.5,1X)/))
  570 FORMAT(17X,F9.5,1X,F9.3)
  950 FORMAT(2X,'    U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
  565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
      DO 1679 J=1,N
      T(1,J)=T(2,J)
      RO(1,J)=1.0/T(1,J)
      U(1,J)=C(J)/RO(1,J)
 1679 CONTINUE
 1697 GO TO 43
 1703 STOP
      END
@
```

APPENDIX F

```
DAY
Tuesday, November 13, 1979 20:28:37
@TY SPD6.FOR
C          SPD6--NONLINEAR TUBE FLOW HEAT TRANSFER PROBLEM,
C             ERROR METHOD,                NOVEMBER 4,1979
          DIMENSION T(2,101),U(2,101),RO(2,101),
         1C(101),A(101),B(101),TR(101),
         3R(101),DEL(101),TO(2,101)
          OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
          READ(1,10) M,N,RE,PR,TW
       10 FORMAT(2I10,10F10.5)
          DR=0.5/FLOAT(N-1)
            DX=DR*DR*RE/2.5
            IP=1
          WRITE(5,15) M,N,DX,RE,PR,TW
       15 FORMAT(2I5,8(F10.5,1X)/)
C         BOUNDARY CONDITIONS
          R(1)=0.0
          N1=N-1
          DO 16 J=1,N1
          J1=J+1
          R(J1)=R(J)+DR
       16 CONTINUE
          COP=8.0*DR**2*RE*PR/DX
          DO 30 J=1,N
          AJ=J-1
          DO 25 I=1,2
          T(I,J)=1.0+(TW-1.0)*(AJ*DR*2.0)**2
          RO(I,J)=1.0/T(I,J)
          U(I,J)=1.0-(2.0*R(J))**2
       25 CONTINUE
          C(J)=U(1,J)*RO(1,J)
       30 CONTINUE
          C(N)=0.0
          U(1,N)=0.0
          U(2,N)=0.0
          IK=1
          SMTUR=0.0
          SMUR=0.0
          DO 33 J=2,N1
          AJ1=J-1
          SMTUR=SMTUR+T(1,J)*U(1,J)*AJ1
          SMUR=SMUR+U(1,J)*AJ1
       33 CONTINUE
          TBBB=SMTUR/SMUR
          QW=260.0*(TW-T(1,N1))/DR
          ANULT=QW/(260.0*(TW-TBBB))
          WRITE(5,535) IK,NIT,(T(1,J),J=1,N)
          WRITE(5,570) ANULT,QW
          WRITE(5,950) (U(1,J),J=1,N)
          WRITE(5,565) X,(RO(1,J),J=1,N)
          X=0.0
       43 IK=IK+1
            IP=IP+1
          IF(IK-M)46,46,1703
       46 X=X+DX
C         UNKNOWS NEAR THE CENTRAL LINE(CENTRAL B.C.)
          NIT=1
          DO 57 I=1,2
          T(I,2)=T(I,1)
          U(I,2)=U(1,1)
          RO(I,2)=RO(1,1)
       57 CONTINUE
```

```
C          UNKNOWNS IN THE FIELD(2,N1)
   61 NT=0
      A(2)=1.0
      B(2)=0.0
      DO 1300 J=2,N1
      J1=J-1
      AJ1=J1
      AJ12=(AJ1+0.5)/AJ1
      AJ32=(AJ1-0.5)/AJ1
      CCOP=COP*C(J)+4.0*T(1,J)
      SQT2=SQRT((0.25*COP*C(J))**2+COP*C(J)*T(1,J)+0.5*AJ12*
     1(T(1,J+1)+T(2,J+1))**2+0.5*AJ32*(T(1,J-1)+T(2,J-1))**2)
      TN(J)=-0.25*CCOP+SQT2
      ROT=TN(J)-T(2,J)
      AROT=ABS(ROT)
      CA=2.0*SQT2/(AJ12*(T(1,J+1)+T(2,J+1)))
      CB=AJ32*(T(1,J-1)+T(2,J-1))/(AJ12*(T(1,J+1)+T(2,J+1)))
      A(J+1)=CA-CB/A(J)
      B(J+1)=CB*B(J)/A(J)-CA*ROT
      IF(AROT-0.000001) 1300,1300,1150
 1150 NT=1
 1300 CONTINUE
      IF(NT.LE.0) GO TO 1567
      DEL(N1)=-B(N)/A(N)
      T(2,N1)=T(2,N1)+DEL(N1)
      DO 1313 J=2,N1
      NJ=N-J
      NJ1=N+1-J
      DEL(NJ)=(DEL(NJ1)-B(NJ1))/A(NJ1)
      T(2,NJ)=T(2,NJ)+DEL(NJ)
 1313 CONTINUE
      T(2,1)=T(2,2)
      T(2,N)=TW
      NIT=NIT+1
      GO TO 61
 1567 QW=260.0*(TW-T(2,N1))/DR
        DO 919 J=1,N
        RO(2,J)=1.0/T(2,J)
        U(2,J)=C(J)/RO(2,J)
  919   CONTINUE
      SMTUR=0.0
      SMUR=0.0
      DO 908 J=2,N1
      J1=J-1
      SMTUR=SMTUR+T(2,J)*U(2,J)*R(J)*(2.0*DR)
      SMUR=SMUR+U(2,J)*R(J)*(2.0*DR)
  908 CONTINUE
      TBBB=SMTUR/SMUR
      ANULT=QW/(260.0*(TW-TBBB))
        IF(IP.NE.11) GO TO 731
        IP=1
      WRITE(5,535) IK,NIT,(T(2,J),J=1,N)
      WRITE(5,570) ANULT,QW
      WRITE(5,950) (U(2,J),J=1,N)
      WRITE(5,565) X,(RO(2,J),J=1,N)
  535 FORMAT(2X,' T,QW',2I5,5(F9.5,1X)/25
     1  (17X,5(F9.5,1X)/))
  570 FORMAT(17X,F9.5,1X,F9.3)
  950 FORMAT(2X,'     U',10X,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
  565 FORMAT(2X,' X,RO',F10.6,5(F9.5,1X),/25(17X,5(F9.5,1X)/))
  731   DO 1679 J=1,N
      T(1,J)=T(2,J)
      RO(1,J)=1.0/T(1,J)
      U(1,J)=C(J)/RO(1,J)
 1679 CONTINUE
 1697 GO TO 43
```

```
15500      1703 STOP
15600           END
@
```

APPENDIX G

```
DAY
 Tuesday, November 13, 1979 21:14:26
@TY SPD7.F
@TY SPD7.FOR
      C        SPD7--HEAT TRANSFER WITH RADIATION,AN UNSTEADY,
      C        ONE DIMENSIONAL FIN PROBLEM,NOVEMBER 13,1979
      C          SUCCESSIVE METHOD
               DIMENSION T(2,101),D(101),TN(101)
               OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
               READ(1,10) GC,CC,RC,TR,B
               READ(1,20) M,N
         10    FORMAT(8F10.5)
         20    FORMAT(4I10)
               N1=N-1
               DX=GC/FLOAT(N1)
               DT=DX*DX/2.5
               WRITE(5,30) GC,CC,RC,TR,B,DX,DT,M,N
         30    FORMAT(10X,7F10.5,2I10/)
               DO 50 I=1,2
               DO 40 J=1,N
               T(I,J)=1.0
         40    CONTINUE
         50    CONTINUE
               IK=1
               NIT=1
               TIME=0.0
               WRITE(5,53) IK,NIT,TIME,(T(1,J),J=1,N)
         53    FORMAT(2X,2I5,F8.5,1X,5(F9.5,1X)/,20(21X,5(F9.5,1X)/)/)
         56    NIT=1
         58    NT=0
               TN(1)=T(1,1)
               DO 80 J=2,N
               TM=0.5*(T(1,J)+T(2,J))
               D(J)=1.0+RC*(TM**3+4.0*TR*TM**2+6.0*TR**2*TM+4.0*TR**3)
               IF(J.EQ.N) GO TO 87
               UCF=1.0/DX**2-1.0/DT+0.5*D(J)
               ACF=1.0/DX**2+1.0/DT+0.5*D(J)
               CTG=T(2,J+1)+T(2,J-1)+T(1,J+1)+T(1,J-1)
               TN(J)=0.5*CTG/(DX**2*ACF)-UCF*T(1,J)/ACF
               ROT=TN(J)-T(2,J)
               AROT=ABS(ROT)
               IF(AROT.LE.0.000001) GO TO 77
               NT=1
         77    IF(J.NE.N1) GO TO 80
               DNP=0.25*D(J)
               T1P=0.25*T(1,J)
         80    CONTINUE
         87    DNP=DNP+0.75*D(N)
               T1P=T1P+0.75*T(1,N)
               TM1=0.5*(T(1,N1)+T(2,N1))
               TUP=(TM1-0.5*T(1,N))/DX-0.5*DX*(0.25*T(2,N1)-T1P)/DT
            1  -CC*(0.5*D(N)*T(1,N)+DX*DNP*(0.25*TM1+0.375*T(1,N))/B)
               TLP=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DX*DNP/B)
               TN(N)=TUP/TLP
               DO 110 J=1,N
               T(2,J)=TN(J)
        110    CONTINUE
               IF(NT.EQ.0) GO TO 145
               NIT=NIT+1
               GO TO 58
        145    IK=IK+1
               TIME=TIME+DT
               WRITE(5,53) IK,NIT,TIME,(T(2,J),J=1,N)
```

```
      DO 153 J=1,N
      T(1,J)=T(2,J)
  153 CONTINUE
      IF(IK.LT.M) GO TO 56
      STOP
      END
@
```

APPENDIX H

```
DAY
Tuesday, November 13, 1979 21:32:51
@TY SPD8.FOR
C          SPD8--HEAT TRANSFER WITH RADIATION,AN UNSTEADY,
C          ONE DIMENSIONAL FIN PROBLEM,NOVEMBER 13,1979
C            ERROR METHOD
           DIMENSION T(2,101),D(101),TN(101),AT(101),BT(101),DEL(10
1)
           OPEN(UNIT=1,ACCESS='SEQIN',DIALOGUE)
           READ(1,10) GC,CC,RC,TR,B
           READ(1,20) M,N
     10    FORMAT(8F10.5)
     20    FORMAT(4I10)
           N1=N-1
           DX=GC/FLOAT(N1)
           DT=DX*DX/2.5
           WRITE(5,30) GC,CC,RC,TR,B,DX,DT,M,N
     30    FORMAT(10X,7F10.5,2I10/)
           DO 50 I=1,2
           DO 40 J=1,N
           T(I,J)=1.0
     40    CONTINUE
     50    CONTINUE
           IK=1
           IP=1
           NIT=1
           TIME=0.0
           WRITE(5,53) IK,NIT,TIME,(T(1,J),J=1,N)
     53    FORMAT(2X,2I5,F8.5,1X,5(F9.5,1X)/,20(21X,5(F9.5,1X)/)/)
     56    NIT=1
     58    NT=0
           AT(2)=1.0
           BT(2)=0.0
           DO 80 J=2,N
           TM=0.5*(T(1,J)+T(2,J))
           D(J)=1.0+RC*(TM**3+4.0*TR*TM**2+6.0*TR**2*TM+4.0*TR**3)
           CDJ=RC*(1.5*TM**2+4.0*TR*TM+3.0*TR**2)
           IF(J.EQ.N) GO TO 87
           UCF=1.0/DX**2-1.0/DT+0.5*D(J)
           ACF=1.0/DX**2+1.0/DT+0.5*D(J)
           CTG=T(2,J+1)+T(2,J-1)+T(1,J+1)+T(1,J-1)
           TN(J)=0.5*CTG/(DX**2*ACF)-UCF*T(1,J)/ACF
           ROT=TN(J)-T(2,J)
           AROT=ABS(ROT)
           CA=2.0*DX**2*ACF+0.5*(DT*CTG+4.0*DX**2)*CDJ
         1 /(DT*ACF)
           CB=2.0*DX**2*ACF
           AT(J+1)=CA-1.0/AT(J)
           BT(J+1)=BT(J)/AT(J)-CB*ROT
           IF(AROT.LE.0.000001) GO TO 77
           NT=1
     77    IF(J.NE.N1) GO TO 80
           DNP=0.25*D(J)
           CDN1=CDJ
           T1NP=0.25*T(1,J)
     80    CONTINUE
     87    IF(NT.EQ.0) GO TO 145
           DNP=DNP+0.75*D(N)
           T1NP=T1NP+0.75*T(1,N)
           TM1=0.5*(T(1,N1)+T(2,N1))
           TUN=(TM1-0.5*T(1,N))/DX-0.5*DX*(0.25*T(2,N1)-T1NP)/DT
         1 -CC*(0.5*D(N)*T(1,N)+DX*DNP*(0.25*TM1+0.375*T(1,N))/B)
```

```
      TLN=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DNP*DX/B)
      ROT=TUN/TLN-T(2,N)
      SGRU=0.5/DX-0.125*DX/DT-0.25*CC*DX*(0.375*T(1,N)*CDN1
     1+0.5*DNP+0.125*(T(2,N1)+T(1,N1))*CDN1+0.375*T(2,N)*CDN1)
/B
      SGRD=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DX*DNP/B
     1+(0.5+0.28125*DX/B)*T(2,N)*CDJ+(0.5*T(1,N)+0.28125*
     2DX*T(1,N)/B+0.09375*DX*(T(2,N1)+T(1,N1))/B)*CDJ)
      SGR=SGRU/SGRD
      PRL=0.5/DX+0.375*DX/DT+CC*(0.5*D(N)+0.375*DNP*DX/B)
      DEL(N1)=(-BT(N)+ROT*PRL/SGRD)/(-SGR+AT(N))
      DEL(N)=AT(N)*DEL(N1)+BT(N)
      N2=N1-1
      DO 93 J=2,N2
      JR=N-J
      JR1=N+1-J
      DEL(JR)=(DEL(JR1)-BT(JR1))/AT(JR1)
   93 CONTINUE
      DO 110 J=2,N
      T(2,J)=T(2,J)+DEL(J)
  110 CONTINUE
      NIT=NIT+1
      GO TO 58
  145 IK=IK+1
      TIME=TIME+DT
      IP=IP+1
      IF(IP.NE.11) GO TO 149
      IP=1
      WRITE(5,53) IK,NIT,TIME,(T(2,J),J=1,N)
  149 DO 153 J=1,N
      T(1,J)=T(2,J)
  153 CONTINUE
      IF(IK.LT.M) GO TO 56
      STOP
      END
@
```

```
 LOGO
Killed Job 38, User F.1SHEN, Account , TTY 36,
  at 13-Nov-79 21:34:06, Used 0:12:23 in 6:11:50
```